游易-程序主题分享合辑

知识管理部 等 2022.06.07 15:37 6236 321 184个资源

汇总从101至今的游易征稿文章合辑。

G107的迷雾方案

陈科锡 2020.04.08 18:51 1321 31 6 查看原文

> 本文描述了一种，按不规则区域开放的迷雾方案，在不使用大面积的迷雾遮罩贴图的情况下，使用类似于3d贴图的方案控制雾区域的开方。
> 基于UE4后处理体来实现，记录了使用UE4后处理上踩到的一些坑，和解决过程。

我们是slg游戏，游戏区域划从大到小分为：行省--》城市--》地块。策划需求是，迷雾的关闭与开放精确到地块。效果如下图：

实现思路

我们使用的是ue4引擎，使用的是一个unbound的后处理体。要解决的问题，无非就是在后处理中，对迷雾区域的计算。

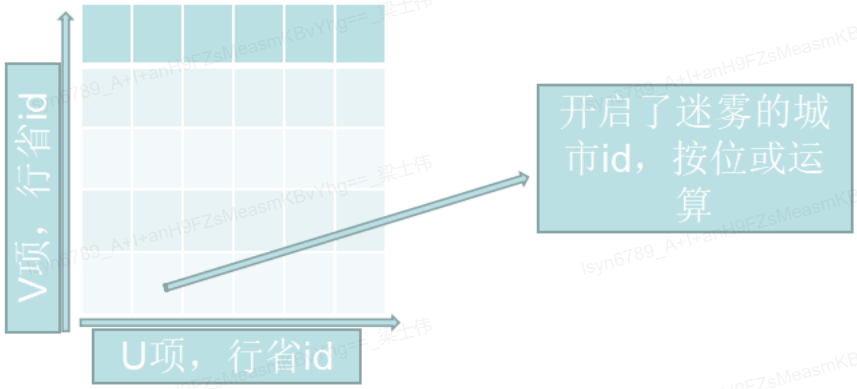
如效果图所见，我们的地块的分布全是不规则的形状，而且我们的大世界目前已经有240000X240000的大小。无论是迷雾区域法，还是遮罩贴图的大小，对性能都会有一定的消耗。所以，我们一开始就否决掉了，计算场景的迷雾遮罩贴图的方案。

最后，跟据曾哥讨论完后，跟据曾哥的建议，我们采用了如下的解决方案：
1、 使用一张64X64的贴图来做为迷雾贴图，用法有点类似3d贴图。其中，U项为行省id， 因为我们的游戏中的行省数量不会超V项为块id，因为我们每个城市的块数量也不会超过64。

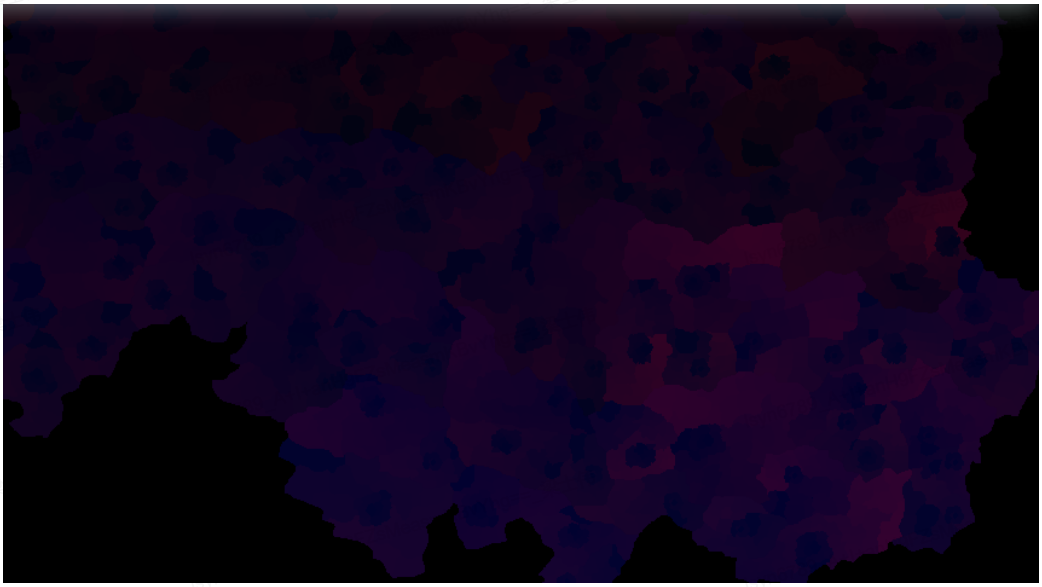
每个像素存下的值，为开启了迷雾的城市的并值（采用按位或操作来运算）。

举个栗子：
当收到服务器的协议，开启，2号行省，1号城市，2号地块的迷雾时。就会在迷雾贴图的（2，2）像素里，填充（1<<1）

当再收到服务器的协议，开启，2号行省，2号城市，2号地块的迷雾时。就会在迷雾贴图的（2，2）像素里，填充（1<<1）|（1<

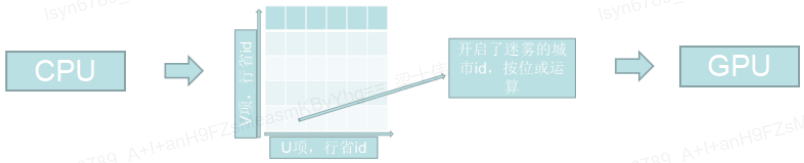


2、我们游戏中有一张全场景有的行省分布mask图，用worldpos从该mask图中采样得到的像素值，包含了这个worldpos所属城市，和地块。



在后处理中，我们遂像素去求出worldpos，然后再使用worldpos.x, world.pos.y到上面的这个mask图中采样，得到了该worldpos
省，城市，和地块id.

3、有了行省，城市，和地块id之后，我们用行省id 和 地块id 去，第1步中的迷雾贴图里采样。采到的值，再与城市id做与运行
零的值，即表示开启了迷雾。



如此，我们便能在后处理中，得到迷雾的区域，效果如下:

可以看到，开了迷雾的区域，锯齿非常严重。由于我们的地块形状都是不规则的，实在想不到什么好的过渡算法，使用迷雾区域变，

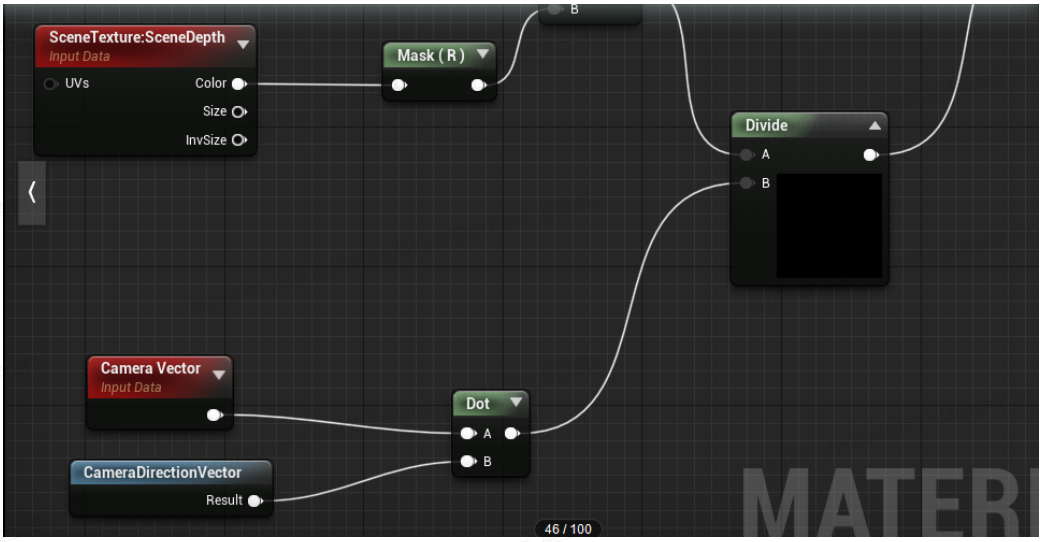
所以，最后解决锯齿的方案，应该大家也已经想到了。我们是多加了两个pass，先将当前屏幕的迷雾区域渲到一张rt中，再对这blur，最后再与scene color做叠加输出。

2 UE后处理使用过程中，踩过的一些坑：

1、关于后处理的quad中，遂像素求world pos。

基于牛叉的经验，认为(u, v, z)去乘vp矩阵的逆矩阵比较耗。想当然地使用相机位置+相机到quad的像素深度值的方式来求world

但是在后处理材质中使用这些结点：camevector * scenedepth + camera position却效果不对。因为scenedepth是正交变换的，以cameravector和cameradirectionvector的点积才能得到正确的world pos。



但是这种计算方式，在pc上能计算正确，在手机上却不行。因为我创建的后处理材质中的blendable location选项是ue4的默认选
是after tonemapping。导至了最后得到的深度贴图的值不对。所以，还需要将blendable location选项设为before tonemapping。

所以在使用后处理材质的时候，还是得小心考虑一下，最后在ps中使用到的所有色值，受不受tonemapping影响之类的。

如果选用的是before tonemapping，甚至还可以在材质中直接使用absolute world position结点。

2、精度溢出的问题

在一些安卓机上，求world pos不可避免有精度不足的问题，，况且我们的场景大小是240000米的数值。我们最后选用的解决方案
world pos压到 view空间，求view空间内的world pos.

3 后续想法

1、精度溢出的问题，虽然是压到了view空间来计算，但是当相机移动时，得实时传输一些uniform，而且渲染帧与逻辑帧有几帧
导至相机位伸时，迷雾的边延有一定的抖动。后面，或许使用global shader，跟据项目的场影情况和玩法逻辑，来压小world po
间。以减少抖动的问题。

2、在求world pos时，无论是camera direction vector ，还是 camera relative world postion，还是absolute world position， 在
处理代码中效率并不高，当使用了这些结点之后，会传输些许的uniform和做不少的矩阵相乘。或许，关于world pos可以尝试自
计算好camera vector ，再varying到ps中，再加上camera pos来求得。当然，最后还是得再看看最终生成的shader代码是否比
camera direction vector ，camera relative world postion，absolute world position 等结点高效。

4 特别鸣谢:

G107|程序|曾鹃程|Kent(gzzengkuncheng@corp.netease.com) 做这个东西的时候，恰逢项目结点，查bug焦头烂额的同时也有
求要撸。感谢曾哥同学几次在危难之际伸出救命之手。

还可以输入

(可添加1个视频+5张图片) 匿名

最热 最新

FLA(范羽)
6楼 tql
2020-04-23 15:06 回

匿名
5楼 rgba一共32位，每个像素不是只可以表达32个城市的数据吗，为什么可以表达64个
2020-04-13 15:27 回

杨念健
4楼 向大佬学习！！
2020-04-09 11:39 回

KO(田波)
3楼 犀利
2020-04-09 10:52 回

潘志豪
2楼 使用worldpos.x, worldpos.y到上面的这个mask图中采样是什么意思？worldpos.x, worldpos.y除以24000然后去采样吗？k图有多大？
2020-04-09 10:17 回

陈道长(陈科锡) 私聊?
2020-04-10 12:04 作者回复 回复

CD(黄尧)
1楼 学习了
2020-04-09 10:13 回

加载完毕,没有更多了

技术 129-工具的设计 128-游戏中的相机 127-游戏中的后处理 126-游戏中的水体 125-手游内存优化 124-AOI可见性&阻... 123-技术助力长线